CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Sequential versus Integrated Optimization: Production, Location, Inventory Control and Distribution

Maryam Darvish
Leandro C. Coelho

July 2017

CIRRELT-2017-39

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2017-005.

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval McGill Concordia University ÉTS UQÀM Université du Québec à Montréal HEC MONTRÉAL Polytechnique Montréal Université de Montréal

Sequential versus Integrated Optimization: Lot Sizing, Inventory Control and Distribution[ǂ]

Maryam Darvish[*], Leandro C. Coelho

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, 2325 de la Terrasse, Université Laval, Québec, Canada G1V 0A6

Abstract. Traditionally, a typical approach towards supply chain planning has been the sequential one. Ignoring the links between decisions, this approach leads to each department of a company making its own decisions, regardless of what others are doing, and overlooking the synergy of a global strategy. However, companies are realizing that significant improvements can occur by exploiting an integrated approach, where various decisions are simultaneously taken into consideration and jointly optimized. Motivated by a real case, in this paper, we consider a production-distribution system that deals with location, production, inventory, and distribution decisions. Multiple products are produced in a number of plants, transferred to distribution centers, and finally shipped to customers. The objective is to minimize total costs while satisfying demands within a delivery time window. We first describe and model the problem and then solve it, using both sequential and integrated approaches. To solve the problem sequentially, we exploit three commonly used procedures based on separately solving each part of the problem. The integrated problem is solved by both an exact method and a matheuristic approach. Our extensive computational experiments and analysis compare solution costs obtained from the two approaches, highlight the value of an integrated approach, and provide interesting managerial insights.

Keywords. Logistics, integrated optimization, sequential decision making, delivery time window, location analysis.

Acknowledgements. This project was partly funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2014-05764. This support is greatly acknowledged. The support rendered by our industrial partner is also highly appreciated.

[ǂ] Revised version of CIRRELT-2016-60.

---

* Corresponding author: Maryam.Darvish@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

1. Introduction

The ultimate goal of any production system is to fulfill the demand of its customers quickly and efficiently. This goal is achieved through effective and efficient supply chain planning. Historically, supply chain planning has been conducted in a sequential or hierarchical fashion. This approach treats each supply chain decision separately from the others. Therefore, in such a disintegrated planning system, even despite the high cost associated with holding stocks, inventory plays an important role in satisfying the demand in a timely manner and linking different functions of the supply chain.

In recent years, the increasing competition among supply chains has forced companies to seek solutions that result in saving cost and improving the efficiency on the one hand, and offering even faster and more flexible service to the customers on the other. Inventory optimization has become the main target for cost reduction initiatives. Recent emphasis on inventory cost reduction coupled with the growing transportation cost and competitive delivery dates accentuate the importance of coordination and integration of supply chain functions and decisions (Fumero and Vercellis, 1999). Under an integrated approach, various functions and decisions within a supply chain are simultaneously treated and jointly optimized. In the sequential approach, typically known as management in silos, the solution obtained from one level is imposed to the next one in the hierarchy of decisions (Vogel et al., 2016). Ignoring the links between decisions, this approach results in sub-optimal solutions. On the contrary, most research and case studies on supply chain integration confirm the positive effect of integration on business performance (Adulyasak et al., 2015; Coelho et al., 2014). Hence, supply chain integration is recognized as the linchpin of success for today's companies (Archetti et al., 2011).

In this paper, we describe, model, and solve a multi-product, multi-plant, multi-period, multi-echelon integrated production, inventory, and distribution problem. This integrated problem has three distinct features of direct shipment, delivery time windows, and dynamic location decisions for distribution centers (DCs).

The transportation decision in integrated production and distribution literature is considered as either direct shipment (full-truck loads) or vehicle routes (milk runs). With the large number of firms outsourcing the transportation function to third party logistics service providers (Amorim et al., 2012), direct shipment is considered in this paper.

Owing to its significant research and practical potential, much attention is devoted to time windows. We consider a delivery time window, meaning that the demand must be satisfied within a specific time frame.

Facility location planning has always been a critical strategic decision. Once the locations are determined, all other decisions such as production quantities, inventories, and transportation can be made. In modern days, customers always impose tighter delivery time windows, therefore, keeping a high service level and managing inventory require simultaneous production and dynamic facility location planning. The integrated production-location problem has become so prevalent that flexible network integration is identified as one of the important recent trends in logistics (Speranza, 2016). Hence, following this trend, in this paper, we study a flexible supply network by considering geographically dispersed DCs available to be rented for a specific period of time.

The objective is to operate a production-distribution system that minimizes production, location, inventory, and distribution costs while satisfying demands within a predetermined delivery time window. To the best of our knowledge, this rich problem has not yet been studied in the literature. Th problem is inspired by a real-world case. Our industrial partner is facing a steady but gradual increase in demand, which requires expanding the operations. To date, the company has invested abundant capital on its production and storage facilities and therefore, production capacities exceed the demand of the company for the moment. However, with the increasing demand growth rate, capacity constraints seem to be fated. Currently, the production manager makes decisions on the production scheduling and quantities, which are later used by the transportation manager to plan the distribution. At this point, the company is interested in how to conduct production planning to save on costs, but at the same time to maintain a high service level.

To solve the problem in a sequential manner, we exploit three commonly used procedures. These procedures decompose the main problem into easier subproblems and then solve each of them separately. Two of these procedures mimic the current situation in companies while one is a lower bound procedure used as a benchmark. Taking an integrative approach, we solve the problem by both an exact method and a matheuristic. Our matheuristic combines an adaptive large neighborhood search (ALNS) heuristic with an exact method.

In summary, the main contributions of this paper are as follows. First, we describe and model a real-life problem in which production, inventory, distribution, and facility location decisions are simultaneously taken into consideration. Second, sequential and integrated optimization approaches are applied. We exploit an exact and a heuristic method to solve the integrated problem. Finally, we demonstrate the value of the integrated approach by comparing its costs with those obtained from the sequential approach. Moreover, we evaluate the quality and performance of all these methods by comparing them with the solutions obtained from the exact methods.

The remainder of this paper is organized as follows. Section 2 provides an overview of the relevant integrated production-distribution literature. In Section 3, we formally describe and model the problem at hand. This is followed by a description of the procedures used to solve the problem sequentially in Section 4. Our proposed integrated matheuristic is explained in Section 5. We present the results of extensive computational experiments in Section 6, followed by the conclusions in Section 7.

2. Literature review

Despite the abundance of conceptual and empirical studies on supply chain integration and coordination, e.g., Power (2005); Mustafa Kamal and Irani (2014), until recently, integrated models of supply chains have been sparse in the operations research literature. Simultaneous optimization of critical supply chain decisions, by integrating them into a single problem, has been such a complex and difficult task that the common approach to solving any integrated problem was to treat each decision separately. Mainly due to their nature, operational level decisions are the targets for integration, among which production and distribution decisions are the most important ones. Independently, both production and distribution problems have several well-studied variants, and so does their integration. As of now, few reviews on various integrated production-distribution models exist, e.g., Sarmiento and Nagi (1999); Mula et al. (2010); Chen (2010); Fahimnia et al. (2013), and Adulyasak et al. (2015). Focusing on the studies that integrate production with direct shipment, in what follows, we briefly review the relevant literature. A list of these papers with their features is presented in Table 1.

Ekşioğlu et al. (2006) formulate the production and transportation planning problem as a network flow and propose a primal-dual based heuristic to solve it. In their model, plants are multi-functional, production and setup costs vary from one plant to another as well as from

Reference	Number of				Inventory	Setup	Location
	Products	Echelons	Periods	Plants			
Ekşioğlu et al. (2006)	S	S	M	M	P	✓	
Akbalik and Penz (2011)	S	S	M	M	P,C	✓	
Sharkey et al. (2011)	S	S	M	M	P		✓
Jolayemi and Olorunniwo (2004)	S	S	M	M	C	✓	
Darvish et al. (2016)	S	S	M	M	P	✓	✓
Park (2005)	M	S	M	M	P,C	✓	
Ekşioğlu et al. (2007)	M	S	M	M	P	✓	
Melo and Wolsey (2012)	M	S	M	S	C	✓	
Nezhad et al. (2013)	M	S	S	M	–	✓	✓
De Matta et al. (2015)	M	S/M	M	M	DC	✓	
Liang et al. (2015)	M	S	M	M	C	✓	✓
Barbarosoğlu and Özgür (1999)	M	M	M	S	P	✓	
Jayaraman and Pirkul (2001)	M	M	S	M	–	✓	✓
This paper	**M**	**M**	**M**	**M**	**P,DC**	✓	✓

Number of products, echelons, periods and plants: S: Single - M: Multiple

Inventory at: P: Plant - DC: Distribution center - C: Customer

Table 1: Integrated production-distribution problems

one period to the next, and transportation costs are concave. Aiming to compare just-in-time and time window policies, Akbalik and Penz (2011) consider delivery time windows. With the just-in-time policy, customers receive a fixed amount whereas, with the time window policy the deliveries are constrained by the time windows. In their model, costs change over time and a fixed transportation cost per vehicle is assumed. A dynamic programming (DP) algorithm is used to solve the problem. The results show that the time window policy has lower cost than the just-in-time one, furthermore, by comparing the mixed integer linear programming (MILP) and DP methods, the authors show that even for large size instances the DP outperforms the MILP. Sharkey et al. (2011) apply a branch-and-price method for an integration of location and production planning in a single sourcing model. The findings show the potential benefits of integrating facility location decisions with the production planning. The proposed branch-and-price algorithm works better when the ratio of the number of customers to the number of plants is low. Darvish et al. (2016) investigate a rich integrated capacitated lot sizing problem (LSP) with a single-product, multi-plant, and multi-period setting. They incorporate direct shipment, delivery time windows, and facility location decisions. They use a branch-and-bound approach to solve the problem. Assessing the trade-offs between costs and fast deliveries, they show the competitive advantage of the integrated approach, both in terms of total costs and service level.

In the profit maximization model presented by Jolayemi and Olorunniwo (2004), any shortfall in demand can be overcome by either increasing capacity or subcontracting. They introduce a procedure to reduce the size of the zero-one MILP and, using a numerical example, they show that the reduced and full-size models generate exactly the same results. Another paper with a profit maximization objective function is that of Park (2005). The model allows stockout and uses homogeneous vehicles for direct shipments. They develop a two-phase heuristic; in the first phase, the production and distribution plans are identified while in the second, the plans are improved by consolidating the deliveries into full truckloads. Only for the small instances, the heuristic generates good results. The paper also investigates the benefits of the integrated approach compared to the decoupled planning procedure, concluding that with the integrated approach both the profit and the demand fill rate increase. Ekşioğlu et al. (2007) extend the problem studied in Ekşioğlu et al. (2006) by considering multiple products. They apply a Lagrangian decomposition heuristic to solve the problem. The problem investigated by Melo and Wolsey (2012) is similar to that of Park (2005). They develop formulations and heuristics that yield solutions with 10% gap for instances with limited transportation capacity but up to 40% for instances with joint production/storage capacity restrictions. Nezhad et al. (2013) tackle an integration of location, production with setup costs, and distribution decisions. In their problem plants are single-source and not capacitated. They propose Lagrangian-based heuristics to solve the problem. The integrated production-distribution problem addressed in De Matta et al. (2015) assumes that each plant uses either direct shipment or a consolidated delivery mode provided by a third party logistics firm. They use Benders decomposition to select the delivery mode and to simultaneously schedule the production. Liang et al. (2015) allow backlogging in the model and propose a hybrid column generation and relax-and-fix method, the exact approach provides the lower bounds while the decomposition yields the upper bounds.

In Barbarosoğlu and Özgür (1999), a Lagrangian-based heuristic is applied to solve an integrated production-distribution problem. They propose a decomposition technique to divide the problem into two subproblems and to optimize each of them separately. Jayaraman and Pirkul (2001) incorporate procurement of the raw material and supply side decisions into the model. Generating several instances, first, they compare the bounds from the Lagrangian approach with the optimal solution obtained by a commercial solver. Then, they apply the proposed method to the data obtained from a real case.

3. Problem description and mathematical formulation

We now formally describe the integrated production, facility location, inventory management, and distribution with delivery time windows problem. We consider a set of plants, available over a finite time horizon, producing multiple products. Starting a new lot incur a setup cost at each plant where a variable cost proportional to the quantity produced is also considered. Each plant owns a warehouse where the products are stored. An inventory holding cost is due for the products kept at these warehouses. The products are then sent to DCs, to be shipped to the final customers. There is a set of potential DCs from which some are selected to be rented. A fixed cost is due and the DC remains rented for a given duration of time. DCs charge an inventory holding cost per unit per period. The products are finally shipped to the geographically scattered customers to satisfy their demand. There is a maximum allowed lateness for the delivery of these products to customers, meaning that the demand must be met within the predetermined delivery time window. A service provider is in charge of all shipments, from plants to DCs and from DCs to final customers. The transportation cost is proportional to the distance, the load, and the type of product being shipped.

Formally, the problem is defined on a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N} = \{1, \ldots, n\}$ is the node set and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$ is the arc set. The node set $\mathcal{N}$ is partitioned into a plant set $\mathcal{N}_p$, a DC set $\mathcal{N}_d$, and a customer set $\mathcal{N}_c$, such that $\mathcal{N} = \mathcal{N}_p \cup \mathcal{N}_d \cup \mathcal{N}_c$. Let $\mathcal{P}$ be the set of P products, and $\mathcal{T}$ be the set of discrete periods of the planning horizon of length T. The inventory holding cost of product p at node $i \in \mathcal{N}_p \cup \mathcal{N}_d$ is denoted as h_{pi}, the unit shipping cost of product p from the plant i to the DC j is c_{pij}, and the unit shipping cost of product p from the DC j to the customer k is c'_{pjk}. Let also f_i be the fixed rental fee for DC i; once selected, the DC will remain rented for the next g periods. Let s_{pi} be the fixed setup cost per period for product p in plant i, v_{pi} be the variable production cost of product p at plant i, and d^t_{pi} be the demand of customer i for product p in period t. The demand occurring in period t must be fulfilled until period $t + r$, as r represents the delivery time window. For ease of representation let D be the total demand for all products from all customers in all periods, i.e., $D = \sum\limits_{t \in \mathcal{T}} \sum\limits_{p \in \mathcal{P}} \sum\limits_{i \in \mathcal{N}_c} d^t_{pi}$.

To solve this rich integrated problem, in each period of the planning horizon, one needs to determine: the product(s) and quantities to be produced in each plant, the DCs to be selected, the inventory levels in plants and DCs, the quantity of products sent from plants to DCs, if the demand of customer is satisfied or delayed, and the quantity of products sent from DCs to customers.

We formulate the problem with the following binary variables. Let θ_{pi}^{t} be equal to one if product p is produced at plant i in period t, and zero otherwise; λ_i^t be equal to one if and only if DC i is chosen to be rented in period t, to be used for g consecutive periods, and ω_i^t be equal to one to indicate whether DC i in period t is in its leasing period. Integer variables to represent quantities produced and shipped are defined as follows. Let $\alpha_{pij}^{tt'}$ be the quantity of product p delivered from DC i to customer j in period t to satisfy the demand of period t', with $t \geq t'$, ρ_{pi}^{t} represent the quantity of product p produced at plant i in period t, β_{pij}^{t} represent the quantity of product p delivered from plant i to DC j in period t, κ_{pi}^{t} as the amount of product p held in inventory at DC i at the end of period t, and μ_{pi}^{t}, the amount of product p held in inventory at plant i at the end of period t.

Table 2 summarizes the notation used in our model.

Table 2: Notation used in the model

Parameters	
h_{pi}	inventory holding cost of product p at node $i \in \mathcal{N}_p \cup \mathcal{N}_d$
c_{pij}	unit shipping cost of product p from plant i to DC j
c'_{pjk}	unit shipping cost of product p from DC j to customer k
f_i	fixed renting cost for DC i
s_{pi}	fixed setup cost per period for product p in plant i
v_{pi}	variable production cost of product p in plant i
d^t_{pi}	demand of customer i for product p in period t
Sets	
$\mathcal{N}_c$	Set of customers
$\mathcal{N}_p$	Set of plants
$\mathcal{N}_d$	Set of DCs
$\mathcal{T}$	Set of periods
$\mathcal{P}$	Set of products
Variables	
θ^t_{pi}	equals to one if product p is produced at plant i in period t
λ^t_i	equals to one if DC i is chosen in period t to be used for g consecutive periods
ω^t_i	equals to one to indicate whether DC i in period t is in its leasing period
$\alpha^{tt'}_{pij}$	quantity of product p delivered from DC i to customer j in period t, to satisfy the demand of period t'
ρ^t_{pi}	quantity of product p produced at plant i in period t
β^t_{pij}	quantity of product p delivered from plant i to DC j in period t
κ^t_{pi}	amount of product p held in inventory at DC i at the end of period t
μ^t_{pi}	amount of product p held in inventory at plant i at the end of period t
Indices	
p,	Product index
t', t	Period index
i, j	Node index

The problem is then formulated as follows:

$$\min \sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{N}_p}\sum_{t\in\mathcal{T}} v_{pi}\rho^t_{pi} + \sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{N}_p}\sum_{t\in\mathcal{T}} s_{pi}\theta^t_{pi} + \sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{N}_d}\sum_{t\in\mathcal{T}} h_{pi}\kappa^t_{pi} + \sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{N}_p}\sum_{t\in\mathcal{T}} h_{pi}\mu^t_{pi} + \sum_{i\in\mathcal{N}_d}\sum_{t\in\mathcal{T}} f_i\lambda^t_i + \sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{N}_p}\sum_{j\in\mathcal{N}_d}\sum_{t\in\mathcal{T}} c_{pij}\beta^t_{pij} + \sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{N}_d}\sum_{j\in\mathcal{N}_c}\sum_{t\in\mathcal{T}}\sum_{t'\in\mathcal{T}} c'_{pij}\alpha^{tt'}_{pij} \tag{1}$$

subject to:

$$\rho^t_{pi} \quad \leq \quad \theta^t_{pi} D \quad i \in \mathcal{N}_p, t \in \mathcal{T}, p \in \mathcal{P} \tag{2}$$

$$\mu_{pi}^{t} = \mu_{pi}^{t-1} + \rho_{pi}^{t} - \sum_{j\in\mathcal{N}_d} \beta_{pij}^{t} \quad p\in\mathcal{P}, i\in\mathcal{N}_p, t\in\mathcal{T}\setminus\{1\} \tag{3}$$

$$\mu_{pi}^{1} = \rho_{pi}^{1} - \sum_{j\in\mathcal{N}_d} \beta_{pij}^{1} \quad p\in\mathcal{P}, i\in\mathcal{N}_p \tag{4}$$

$$\kappa_{pi}^{t} = \kappa_{pi}^{t-1} + \sum_{j\in\mathcal{N}_p} \beta_{pji}^{t} - \sum_{j\in\mathcal{N}_c}\sum_{t'=t-r}^{t} \alpha_{pij}^{tt'} \quad p\in\mathcal{P}, i\in\mathcal{N}_d, t\in\mathcal{T}\setminus\{1\} \tag{5}$$

$$\kappa_{pi}^{1} = \sum_{j\in\mathcal{N}_p} \beta_{pji}^{1} - \sum_{j\in\mathcal{N}_c} \alpha_{pij}^{11} \quad p\in\mathcal{P} \quad i\in\mathcal{N}_d \tag{6}$$

$$\sum_{p\in\mathcal{P}} \kappa_{pi}^{t} \leq \omega_i^t D \quad i\in\mathcal{N}_d \quad t\in\mathcal{T} \tag{7}$$

$$\sum_{p\in\mathcal{P}} \kappa_{pi}^{t} \leq \omega_i^{t+1} D \quad i\in\mathcal{N}_d \quad t\in\mathcal{T}\setminus\{T\} \tag{8}$$

$$\sum_{\substack{t'=t-g+1\\ t'\geq 1}}^{t} \lambda_i^{t'} \geq \omega_i^t \quad i\in\mathcal{N}_d \quad t\in\mathcal{T} \tag{9}$$

$$\sum_{t'=t}^{t'=t+g-1,\ t'\leq T} \omega_i^{t'} \geq \lambda_i^t min(g, T-t) \quad i\in\mathcal{N}_d \quad t\in\mathcal{T} \tag{10}$$

$$\sum_{t'=t}^{t'=t+g-1,\ t'\leq T} \lambda_i^{t'} \leq 1 \quad i\in\mathcal{N}_d \quad t\in\mathcal{T} \tag{11}$$

$$\sum_{i\in\mathcal{N}_d}\sum_{t=1}^{s}\sum_{t'=1}^{t} \alpha_{pij}^{tt'} \leq \sum_{t=1}^{s} d_{pj}^t \quad p\in\mathcal{P} \quad j\in\mathcal{N}_c \quad s\in\mathcal{T} \tag{12}$$

$$\alpha_{pij}^{tt'} = 0 \qquad p\in\mathcal{P} \qquad i\in\mathcal{N}_d \quad j\in\mathcal{N}_c \quad t\in\mathcal{T} \quad t'\in\{0,...,t-r\}\cup\{t,...,T\} \tag{13}$$

$$\sum_{i\in\mathcal{N}_d}\sum_{t=1,t<T}^{s+r}\sum_{t'=1}^{s} \alpha_{pij}^{tt'} \geq \sum_{t=1}^{s} d_{pj}^t \quad p\in\mathcal{P} \quad j\in\mathcal{N}_c \quad s\in\mathcal{T} \tag{14}$$

$$\sum_{p\in\mathcal{P}}\sum_{j\in\mathcal{N}_c}\sum_{t'=1}^{t} \alpha_{pij}^{tt'} \leq D\omega_i^t \quad i\in\mathcal{N}_d \quad t\in\mathcal{T} \tag{15}$$

$$\sum_{i\in\mathcal{N}_d}\sum_{t\in\mathcal{T}} \alpha_{pij}^{tt'} = d_{pj}^{t'} \quad p\in\mathcal{P} \quad j\in\mathcal{N}_c \quad t'\in\mathcal{T} \tag{16}$$

$$\omega_i^t, \theta_{pi}^t, \lambda_i^t \in \{0,1\} \tag{17}$$

$$\rho_{pij}^{tt'}, \kappa_{pi}^t, \alpha_{ijp}^{tt'}, \beta_{ijp}^t \in \mathbb{Z}^*. \tag{18}$$

The objective function (1) minimizes the total cost consisting of the production setup and variable costs, inventory holding costs, rental fees, and transportation costs, from plants to DCs and also from DCs to final customers. Constraints (2) guarantee that only products set up for production are produced. Constraints (3) and (4) ensure the inventory conservation at each plant. Similarly, constraints (5) and (6) are applied to DCs. Constraints (7) and (8) guarantee that the remaining inventory at the DC is transferred to the next period only if the DC is rented in the next period. Constraints (9)–(11) ensure that once a DC is selected, it will remain rented for the next g consecutive periods. Constraints (11) make sure that the rental fee for each g period is paid only once. Constraints (12) and (13) guarantee that no demand is satisfied in advance, while constraints (14) impose r periods as the maximum allowed lateness for fulfilling the demand. Thus, the total demand up to period t must be delivered by period $t + r$. No delivery to customers can take place from a DC if it is not rented, as ensured by constraints (15). Constraints (16) make sure that every single demand is delivered to the customers. Finally, constraints (17) and (18) define the domain and nature of the variables.

4. Sequential and lower bound procedures

In this section we propose three sequential procedures to solve the problem. Their motivation is twofold. First, we want to mimic production systems managed in silos, as inspired and currently conducted by our industrial partner. Second, we want to assess how a sequential algorithm performs compared to the integrated one proposed in this paper. These comparisons are presented in Section 6. In what follows, in Section 4.1 we present a Top-down procedure, for the cases in which production is the most important part of the process and has priority in determining how the system works. This decision is then followed by inventory allocation to DCs and finally by distribution decisions. In Section 4.2 we describe a Bottom-up procedure, simulating the alternative scenario in which distribution has priority. The distribution decisions are followed by DC allocation, and lastly by production decisions. Finally, in Section 4.3 we describe an Equal power procedure, in which all three departments would have similar positions in the hierarchy of power; we explain how this procedure yields a lower bound on the optimal cost.

4.1. Top-down procedure

In the Top-down procedure, production managers have the most power and therefore, they can determine how the rest of the system works. This method, which observed as the current practice of our industrial partner, works as follows.

First, minimize only production costs $\sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{N}_p} \sum_{t \in \mathcal{T}} v_{pi} \rho_{pi}^t + \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{N}_p} \sum_{t \in \mathcal{T}} s_{pi} \theta_{pi}^t$ subject to (2)–(18). An optimal solution to this problem determines the best production plan without any interaction with downstream decisions. Let these optimal decision values be $\bar{\theta}_{pi}^t$ and $\bar{\rho}_{pi}^t$. Note that because these production decisions are made considering the whole feasible region, determined by (2)–(18), feasibility is ensured.

The second phase works by considering a minimization objective function consisting of only DC-related terms, namely $\sum_{i \in \mathcal{N}_d} \sum_{t \in \mathcal{T}} f_i \lambda_i^t$, subject to (2)–(18), and to $\theta_{pi}^t = \bar{\theta}_{pi}^t$ and $\rho_{pi}^t = \bar{\rho}_{pi}^t$. In this problem, inventory allocation decisions are made subject to the feasible region of the overall problem and the production decisions that had priority over the inventory ones. Let the value of these decision variables be $\bar{\lambda}_i^t$ and $\bar{\omega}_i^t$.

The final phase consists of determining the best way to distribute the inventory to the customers, given fixed production and allocation plans. This is accomplished by minimizing $\sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{N}_d} \sum_{j \in \mathcal{N}_c} \sum_{t \in \mathcal{T}} \sum_{t' \in \mathcal{T}} c'_{pij} \alpha_{pij}^{tt'}$, and subject to the feasible region of the original problem (2)–(18), and to $\theta_{pi}^t = \bar{\theta}_{pi}^t$, $\rho_{pi}^t = \bar{\rho}_{pi}^t$, $\lambda_i^t = \bar{\lambda}_i^t$ and $\omega_i^t = \bar{\omega}_i^t$. By putting together all three levels of decisions, one can obtain the overall solution and easily compute the cost of the solution yielded by the Top-down procedure. A pseudocode of this procedure is presented in Algorithm 1.

Algorithm 1 Top-down procedure

1: Consider all constraints of the problem formulation from Section 3, (2)–(18).
2: Build an objective function with production variables θ_{pi}^t and ρ_{pi}^t:

$$\sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{N}_p} \sum_{t \in \mathcal{T}} v_{pi} \rho_{pi}^t + \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{N}_p} \sum_{t \in \mathcal{T}} s_{pi} \theta_{pi}^t.$$

3: Optimize the problem, obtain optimal values $\bar{\theta}_{pi}^t$ and $\bar{\rho}_{pi}^t$.
4: Fix θ_{pi}^t and ρ_{pi}^t to their obtained values.
5: Add DC-related variables λ_i^t to the objective function:

$$\sum_{i \in \mathcal{N}_d} \sum_{t \in \mathcal{T}} f_i \lambda_i^t.$$

6: Optimize the problem, obtain optimal values $\bar{\lambda}_i^t$ and $\bar{\omega}_i^t$.
7: Fix variables λ_i^t and ω_i^t to their obtained values.
8: Add all variables to the objective function, as it is defined in (1).
9: Optimize the problem, obtain optimal values for all variables.
10: Return the objective function value.

4.2. Bottom-up procedure

In the Bottom-up procedure, we suppose that the distribution managers have the most power and can, therefore, determine how the rest of the system works. This is done by taking all constraints (2)–(18) into account, but optimizing the objective function only for the distribution variables. Once distribution decisions are made and fixed, inventory allocation decisions, namely when and which DCs to rent, are optimized. As mentioned earlier, feasibility is guaranteed. We now solve the same problem with a new set of fixed decisions (related to distribution), and optimize only DC-related costs. When this part is determined, all the decisions are fixed and no longer change. Finally, once DCs have been selected, and all distribution and DC variables are known, we can optimize the remaining variables of the problem. By putting together all three levels of decisions, one can obtain the overall solution and easily compute the cost of the solution yielded by the Bottom-up procedure. A pseudocode of this procedure is presented in Algorithm 2.

Algorithm 2 Bottom-up procedure

1: Consider all constraints of the problem formulation from Section 3, (2)–(18).
2: Build an objective function with distribution variables $\alpha_{pij}^{tt'}$.
$$\sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{N}_d}\sum_{j\in\mathcal{N}_c}\sum_{t\in\mathcal{T}}\sum_{t'\in\mathcal{T}} c'_{pij}\alpha_{pij}^{tt'}.$$
3: Optimize the problem, obtain optimal values $\bar{\alpha}_{pij}^{tt'}$.
4: Fix $\alpha_{pij}^{tt'}$ to their obtained values.
5: Add DC-related variables λ_i^t to the objective function:
$$\sum_{i\in\mathcal{N}_d}\sum_{t\in\mathcal{T}} f_i\lambda_i^t.$$
6: Optimize the problem, obtain optimal values $\bar{\lambda}_i^t$ and $\bar{\omega}_i^t$.
7: Fix variables λ_i^t and ω_i^t to their obtained values.
8: Add all variables to the objective function, as it is defined in (1).
9: Optimize the problem, obtain optimal values for all variables.
10: Return the objective function value.

4.3. Equal power procedure

In the Equal power procedure, we assume that all three decision levels have equal power. Therefore, information is shared with all departments at the same time but decisions are made in parallel and each department optimizes its own decisions. This procedure will likely yield an infeasible solution since each part of the problem is optimized individually. However, the sum of the costs of all three levels indicates the optimal decision for each level, when the costs of the other levels are not considered. Having all three decision levels put together, if these yield a feasible solution, it is obviously optimal, otherwise, their sum constitutes a valid lower bound on the costs of the problem. Algorithm 3 describes the pseudocode for this procedure.

Algorithm 3 Equal power procedure

Consider all constraints of the problem formulation from Section 3, (2)–(18).
Build an objective function with distribution variables $\alpha_{pij}^{tt'}$.
Optimize the problem, obtain optimal values $\bar{\alpha}_{pij}^{tt'}$, and optimal distribution solution z_c.
Build an objective function with DC-related variables λ_i^t, β_{pij}^t, and κ_{pi}^t.
Optimize the problem, obtain optimal values for $\bar{\lambda}_i^t$, $\bar{\omega}_i^t$, $\bar{\beta}_{pij}^t$, and $\bar{\kappa}_{pi}^t$, and optimal DC solution z_d.
Build an objective function with plant-related variables θ_{pi}^t, ρ_{pi}^t, and μ_{pi}^t.
Optimize the problem, obtain optimal values for $\bar{\theta}_{pi}^t$, $\bar{\rho}_{pi}^t$, and $\bar{\mu}_{pi}^t$, and optimal production solution z_p.
if the combination of all three decisions is feasible **then**
    Return optimal solution and its cost $z^* = z_p + z_d + z_c$.
else
    Return lower bound value $\underline{z} = z_p + z_d + z_c$.
end if

5. Integrated solution algorithm

The problem at hand is reducible to the multi-plant uncapacitated LSP and also the joint-replenishment problem, an extension of the uncapacitated fixed charge network flow. The joint-replenishment problem is known to be NP-hard (Cunha and Melo, 2016), as are most variants of the LSP. Although the uncapcitated LSP is easier to solve, the multi-plant version is still NP-complete (Sambasivan and Schmidt, 2002). As is the case of many other NP-hard problems, exact methods can solve small-size instances to optimality in a reasonable time but to obtain good solutions for larger instances, one must develop ad hoc heuristic algorithms. To solve the problem at hand, we propose a matheuristic based on a hybrid of ALNS and exact methods. The ALNS introduced by Ropke and Pisinger (2006) has shown outstanding results in solving various supply chain problems. ALNS, as a very efficient and flexible algorithm, explores large complex neighborhoods and avoids local optima. Hence, because of its generality and flexibility, it is highly suitable for the problem at hand. Our contribution, however, lies in customizing and applying this method to our problem.

We propose a three-level matheuristic approach in which the problem is divided into two subproblems that are then solved in an iterative manner. In the first level, we apply the ALNS heuristic in order to decide which plants and DCs should be selected, and to determine which products have to be produced in any of the selected plants. Once these decisions are fixed, the problem becomes a minimum cost network flow (MCNF) problem. The MCNF finds a feasible flow with minimum cost on a graph in which a cost is associated to each arc (Goldberg, 1997). Therefore, in the second level, all the other remaining decisions on deliveries from selected plants to rented DCs, and from rented DCs to the customers, as well as the inventory level held at plants and rented DCs are obtained exactly by solving an integer linear programming subproblem. This is done efficiently by exploiting the branch-and-bound algorithm and applying it to the MCNF problem. Finally, if needed and to avoid local optimum solutions, we improve the obtained solution and move it toward the global optimal by solving the model presented in Section 3 with exact methods for a very short period of time. The detailed algorithmic framework is as follows.

- *Initial solution*: we start with generating a feasible initial solution by making all plants and DCs selected in all periods. This feasible initial solution is quickly improved by deselecting as many facilities as possible while maintaining feasibility. At this step, costs are not yet of concern and in order to improve the solution, we take all the constraints of (2)–(18) and solve the problem with the following objective function: $\min \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{N}_p} \sum_{t \in \mathcal{T}} \theta^t_{pi} + \sum_{i \in \mathcal{N}_d} \sum_{t \in \mathcal{T}} \lambda^t_i$. We obtain the initial solution s and its corresponding cost $z(s)$ to be improved.

- *Large neighborhood:* at each iteration, one operator from the list described in Section 5.1 is selected. Operators work for any type of facility; therefore, plants and DCs have the same chance of being selected. To diversify the search, each operator is repeated n times, n being drawn from a semi-triangular distribution and bounded between [1,a]. We compute n as in (19) where b is a random number in the [0,1] interval and a is an integer number, starting with a value of one and increasing throughout the iterations.

$$n = \left\lfloor a - \sqrt{(1-b)(a-1)^2} + 0.5 \right\rfloor \tag{19}$$

- *Adaptive search engine:* the operators are selected according to a roulette-wheel mechanism. A weight is associated to previous performances of each of the operators, modulating their chances of being selected.

- *Acceptance criteria:* To diversify the solutions, a simulated annealing-based acceptance rule is applied. The current solution s is accepted over the incumbent solution s' with the probability of $e^{\left(\dfrac{z(s') - z(s)}{H}\right)}$, where H is the current temperature. The temperature is decreased at every iteration by α, where $0 < \alpha < 1$. Once the temperature reaches the final temperature, H_{final}, it is reset to the initial temperature, H_{start}.

- *Adaptive weight adjustment:* A score and a weight are assigned to each of the operators. The weight matrix, which has an initial value of one, is updated at every φ iterations. It is updated using the scores each operator has accumulated. The score matrix is initially set to zero, and the better the operator performs, the higher score it accumulates. We define $\sigma_1 > \sigma_2 > \sigma_3 > 0$. If an operator finds a solution better than the best solution obtained so far, a score of σ_1 will be assigned to it. If the obtained solution by the operator is not the best but it is better than the incumbent solution, the score will be updated by σ_2. Finally, if the solution is no better than the incumbent solution but it is still within the acceptable range, the operator will be given a σ_3 score.

- *Periodic post-optimization:* if no improvement is achieved for more than 2φ iterations, we use the best solution as an input to the model of Section 3 and solve it for 20 seconds with the exact method. If this post-optimization attempt yields an optimal solution, the algorithm stops, as the global optimum has been found; otherwise, if it improves the solution, the improved solution is passed to the ALNS framework and the procedure continues.

- *Stopping criteria:* the algorithm will stop, if either the maximum number of iterations $iter_{max}$ or the maximum allotted time is reached, we limit the running time to one hour. It will also stop, if the solution does not change in more than $\frac{iter_{max}}{2}$ iterations. Moreover, it will stop when the optimal solution is obtained in the periodic post-optimization step.

5.1. List of operators

The operators we have designed to explore the search space with the ALNS framework are as follows.

1. Random: this operator selects a plant, a product and a period (or a DC and a period), and flips its current status; if the facility is not in use it becomes in use, and vice-versa.

2. Based on shipping costs: first, for each product we compare the shipping costs from plants that are not producing any product to all currently rented DCs, and then we identify the combination of product, plant, and period with the lowest cost. The corresponding product is then assigned to be produced at that plant in that period. Similarly, the highest shipping cost induces a product to have its production stopped at the given plant and period.
3. Based on unit costs: among all plants, we identify the plant and the product with the highest unit production cost; we stop production of the identified product in the selected plant; for DCs, we stop renting the one with the highest unit inventory cost.
4. Based on demand: first, we identify the product and period with the highest demand, then we make all the plants produce that product in that period. Similarly, we identify the product and period with the minimum demand, and stop its production in the identified period.
5. Based on delivery quantity to DCs: we identify the plant delivering the least (most) and the DC receiving the least (most) per period. Facilities with the least usage will not be in use; for those with the largest usage, a random DC is rented in the same period, and production for the same plant is set up for all products in the following period.
6. Based on inventory level: we identify the plant and period with the maximum inventory, and ensure it stays in use in that period. If the plant is already in use, we keep it in use also in the next period. For DCs, we stop renting the one with the lowest inventory level during its g leasing periods.
7. Based on production quantity: we identify the product/plant/period combination in which the maximum (minimum) production occurs; we stop production of that product in the plant with the smallest production in the identified period but assign the plant and the product with the maximum production to its next period. We also identify the period with the highest production, and rent an extra DC.
8. Based on delivery quantity to customers: we identify all DC/period combinations with deliveries lower than a percentage of the total demand and among them, we select a DC and end its lease for that period (and consequently the next g periods). Similarly, for plants, we select a random one and stop production of all products in the previously identified period.

5.2. Parameter settings and the pseudocode

We have tested different combinations of parameters and tuned them mainly by trial and error. The initial temperature H_{start} is set to $(r+1) \times 100,000$. This initial temperature is cooled down until it reaches the final temperature $H_{final} = 0.01$. The cooling rate, α, is tuned to 0.999. In our implementation, iteration count is one of the stopping criteria, and it is satisfied once 3,000,000 iterations are performed. We set φ to 1,000 iterations and update the scores with $\sigma_1 = 10$, $\sigma_2 = 4$, and $\sigma_3 = 3$.

The pseudocode for the proposed matheuristic is provided in Algorithm 4.

6. Computational Experiments

We now describe the details related to the computational experiments used to evaluate our algorithms. All computations are conducted on Intel Core i7 processor running at 3.4 GHz with 64 GB of RAM installed, with the Ubuntu Linux operating system. A single thread was used for up to one hour, i.e., a time limit of 3600 seconds was imposed on all algorithms. The algorithms are coded in C++ and we use IBM Concert Technology and CPLEX 12.6.3 as the MIP solver. Section 6.1 describes how the instances are generated, detailed computational results are provided in Section 6.2, and sensitivity analysis and the managerial insights are provided in Section 6.3.

6.1. Generation of the instances

By consultation with our industrial partner, we have generated a large data set by varying the number of products, periods, plants, DCs, and customers. Our test bed is generated as shown in Table 3. The number of plants and DCs are determined by the number of periods: if $T = 5$, then $N_d = 8$ and $N_p = 5$, if $T = 10$, then $N_d = 15$ and $N_p = 10$, and finally if $T = 50$, then $N_d = 25$ and $N_p = 15$. For each of 11 combinations, we generate five random instances. For each instance we consider a delivery time window $r = 0$, 1, 2, or 5 periods. Thus, we solve 220 instances in total.

6.2. Results of the computational experiments

We now present the results of extensive computational experiments carried out to evaluate the performance of all algorithms, and to draw meaningful conclusions for the problem at hand.

Algorithm 4 Proposed matheuristic

Initialize weights to 1, scores to 0, $H \leftarrow H_{start}$.
$s \leftarrow s_{best} \leftarrow$ initial solution.
while stopping criteria are not met **do**
    $s' \leftarrow s$
    Select an operator and apply it to s'
    Solve the remaining flow problem, obtain solution $z(s')$
    if $z(s') < z(s)$ **then**
        if $z(s') < z(s_{best})$ **then**
            $s_{best} \leftarrow s'$
            update the score for the operator used with σ_1
        else
            update the score for the operator used with σ_2
        end if
    else
        if s' is accepted by the simulated annealing criterion **then**
            update the scores for the operator used with σ_3
            $s \leftarrow s'$
        end if
    end if
    $H \leftarrow \alpha \times H$
    if iterations is a multiple of φ **then**
        update weights and reset scores of all operators
        if no improvement found in last 2φ iterations **then**
            if $H < H_{final}$ **then**
                $H \leftarrow H_{start}$
                if no improvement found for $z(s')$ **then**
                    Input s_{best} into the MIP in Section 3 and solve it for 20 seconds
                end if
            end if
        else
            $s \leftarrow s_{best}$
        end if
    end if
end while
Return s_{best}

Table 3: Input parameter values

Name	Parameter	Values
Products	P	1, 5, 10
Periods	T	5, 10, 50
Plants	N_p	5, 10, 15
DCs	N_d	8, 15, 25
Customers	N_c	20, 50, 100
Delivery time window	r	0, 1, 2, 5
DC active period	g	$\frac{T}{5}$
Demand	d_{pk}^t	[0, 2]
Plant setup cost	s_{pi}	[10, 15]
Plant variable cost	v_{pi}	[1, 10]
Fixed DC renting cost	f_j	[100, 150]
Inventory holding cost	h_{pj}	[1, 4]
Shipping cost (plants-DC)	c_{pij}	[10, 100]
Shipping cost (DC-customers)	c'_{pjk}	[10, 1000]

We first describe the results of the experiments with the mathematical model proposed in Section 3. This is followed by the comparison of the performance of the sequential procedures proposed in Section 4, and our integrated hybrid matheuristic from Section 5 with that of the exact algorithms.

Average computational results using the CPLEX branch-and-bound algorithm are presented in Table 4. For each instance, we report the average of the gaps (G) with respect to the lower bound, calculated as $100 \times \dfrac{Upper\ Bound - Lower\ Bound}{Lower\ Bound}$, the number of cases solved to optimality (O), and the average running time (T) in seconds for each predetermined time window r. As presented in Table 4, only the small instances, mostly those with fewer than five products or periods, could be solved to optimality. The parameter controlling the number of periods seems to have a strong effect on the performance of the exact method. Indeed, it has a huge effect on the size of the problem as measured by the number of variables and constraints. Moreover, the length of the delivery time window affects the number of instances solved to optimality, the average gap, and the running time.

To evaluate the performance of the sequential procedures versus our proposed matheuristics

Table 4: Results from the branch-and-bound algorithm

Instance	$r = 0$		$r = 1$		$r = 2$		$r = 5$	
P-T-N_c-N_d-N_p	$G(\%)^{(O)}$	$T(s)$	$G(\%)^{(O)}$	$T(s)$	$G(\%)^{(O)}$	$T(s)$	$G(\%)^{(O)}$	$T(s)$
1-5-20-8-5	$0.00^{(5)}$	2	$0.00^{(5)}$	2	$0.00^{(5)}$	1	$0.00^{(5)}$	0
1-10-100-15-10	$2.43^{(1)}$	3,113	$20.15^{(0)}$	3,606	$25.97^{(0)}$	3,603	$27.63^{(0)}$	3,603
1-10-50-15-10	$0.00^{(5)}$	610	$10.94^{(0)}$	3,616	$18.13^{(0)}$	3,606	$16.50^{(0)}$	3,601
1-50-100-25-15	$30.69^{(0)}$	3,640	$94.85^{(0)}$	3,626	$131.83^{(0)}$	3,624	$247.84^{(0)}$	3,634
5-5-20-8-5	$0.00^{(5)}$	342	$0.13^{(4)}$	1,321	$0.81^{(3)}$	1,703	$0.00^{(5)}$	221
5-10-100-15-10	$14.48^{(0)}$	3,612	$28.41^{(0)}$	3,601	$32.95^{(0)}$	3,601	$39.21^{(0)}$	3,601
5-10-50-15-10	$11.56^{(0)}$	3,610	$22.49^{(0)}$	3,602	$23.19^{(0)}$	3,602	$25.65^{(0)}$	3,602
10-5-20-8-5	$2.08^{(0)}$	3,602	$3.66^{(0)}$	3,604	$4.27^{(0)}$	3,604	$0.05^{(5)}$	1,275
10-10-100-15-10	$14.60^{(0)}$	3,602	$26.93^{(0)}$	3,602	$33.88^{(0)}$	3,610	$34.43^{(0)}$	3,601
10-10-50-15-10	$7.19^{(0)}$	3,605	$21.38^{(0)}$	3,603	$29.74^{(0)}$	3,602	$20.83^{(0)}$	3,611
Average	$8.30^{(0.32)}$	2,574	$22.89^{(0.18)}$	3,018	$30.08^{(0.16)}$	3,056	$41.21^{(0.30)}$	2,675

and to gain insight into managerial decisions related to the problem at hand, we present their results in Tables 5–8, one table per value of the delivery time window r. The improvements with respect to the solution obtained from the exact algorithm by the Top-down, Bottom-up, and matheuristic algorithms are presented along with their running times. For each method, this improvement is obtained as $100 \times \dfrac{Upper\ Bound_{CPLEX} - Cost_{method}}{Upper\ Bound_{CPLEX}}$.

Table 5 presents the results obtained with no delivery time window, i.e., $r = 0$. On average, the proposed method gets slightly better solutions than CPLEX. When only one product is involved, no matter of the number of customers or periods, our proposed method always outperforms CPLEX. Both the Top-down and Bottom-up procedures are very fast, but the costs obtained by these methods are much higher than the ones from CPLEX. As indicated in Table 5, the Bottom-up procedure outperforms the Top-down on almost all large instances with multiple products, more than five periods and 50 customers. Although on average the Bottom-up procedure takes less running time, the results obtained by this procedure are about 1.5 times worse than the ones from the Top-down.

Table 5: Heuristics results for $r = 0$

Instance	Top-down		Bottom-up		proposed method	
P-T-N_c-N_d-N_p1	I (%)	T(s)	I (%)	T(s)	I (%)	T(s)
1-5-20-8-5	−10.03	0	−315.27	0	0.00	1
1-10-100-15-10	−46.41	3	−157.86	2	0.10	3,606
1-10-50-15-10	−33.05	2	−238.78	2	0.00	2,048
1-50-100-25-15	−123.86	558	−123.01	245	8.14	3,602
5-5-20-8-5	−18.42	1	−107.79	0	0.00	1,150
5-10-100-15-10	−119.74	34	−42.70	13	−1.25	3,602
5-10-50-15-10	−70.89	12	−74.98	5	−0.74	3,604
10-5-20-8-5	−32.37	1	−50.13	1	−0.16	3,462
10-10-100-15-10	−179.37	53	−23.22	38	−3.69	3,605
10-10-50-15-10	−116.55	51	−38.67	21	−0.66	3,610
Average	−75.09	72	−117.24	33	0.17	2,829

When $r = 1$, as indicated in Table 6, our approach always outperforms CPLEX, with an average improvement of 4.17%. For a large instance with one product, 50 periods, and 100 customers, this difference is up to 27.03%. Although the solutions obtained by both sequential methods have slightly worsened, the extra delivery period has dramatically increased the running time for the Top-down procedure, with almost no significant effect on the Bottom-up.

Table 6: Heuristics results for $r = 1$

Instance	Top-down		Bottom-up		proposed method	
P-T-N_c-N_d-N_p	I (%)	T(s)	I (%)	T(s)	I (%)	T(s)
1-5-20-8-5	−19.34	0	−423.38	0	0.00	3
1-10-100-15-10	−54.70	17	−201.61	1	2.21	3,608
1-10-50-15-10	−43.01	8	−301.56	3	0.71	3,474
1-50-100-25-15	−84.26	1,391	−162.38	313	27.03	3,601
5-5-20-8-5	−39.68	2	−149.74	0	0.00	2,627
5-10-100-15-10	−142.26	1,012	−68.09	16	2.75	3,419
5-10-50-15-10	−92.443	289	−109.26	6	4.10	3,604
10-5-20-8-5	−54.21	3	−89.97	2	0.08	3,608
10-10-100-15-10	−195.44	1,292	−46.50	38	1.58	3,601
10-10-50-15-10	−129.44	1,054	−60.89	25	3.27	3,604
Average	−85.45	507	−161.34	40	4.17	3,115

Table 7 shows the results obtained by considering two-day delivery time window, i.e., $r = 2$. On average our algorithm improves the solution by 5.29%. As before, the biggest improvement is observed for the large instance with one product, 50 periods and 100 customers, but small instances are either solved to optimality as CPLEX or has been slightly improved. As the time window grows, the performance of both Top-down and Bottom-up procedures declines but compared to the $r = 1$ case, the running time slightly increases.

Table 7: Heuristics results for $r = 2$

Instance	Top-down		Bottom-up		proposed method	
P-T-N_c-N_d-N_p	I (%)	T(s)	I (%)	T(s)	I (%)	T(s)
1-5-20-8-5	−20.41	0	−459.31	0	0.00	4
1-10-100-15-10	−65.78	20	−200.37	3	2.87	3,608
1-10-50-15-10	−51.46	9	−282.99	2	0.57	3,371
1-50-100-25-15	−80.81	1,549	−206.21	317	31.44	3,601
5-5-20-8-5	−45.66	2	−178.08	1	0.04	3,149
5-10-100-15-10	−162.31	1,244	−84.87	24	1.80	3,602
5-10-50-15-10	−102.42	431	−125.29	10	4.87	3,603
10-5-20-8-5	−76.32	3	−109.47	6	0.75	3,606
10-10-100-15-10	−201.60	1,291	−61.37	66	4.74	3,601
10-10-50-15-10	−170.87	1,034	−105.63	24	5.86	3,601
Average	−97.76	558	−181.36	45	5.29	3,174

The best results of our proposed method are obtained for $r = 5$. As presented in Table 8, our method improves the results by 7.62%. The difference in performance of the two methods becomes even more evident for the big instances with 50 periods and 100 customers, in which our proposed method improves the solution obtained by CPLEX up to 49.66%. As before, the two sequential procedures can quickly provide feasible solutions, but of very poor quality.

Table 8: Heuristics results for $r = 5$

Instance	Top-down		Bottom-up		proposed method	
P-T-N_c-N_d-N_p	I (%)	T(s)	I (%)	T(s)	I (%)	T(s)
1-5-20-8-5	−36.13	0	−514.97	0	0.00	2
1-10-100-15-10	−90.91	19	−219.95	4	4.65	3,607
1-10-50-15-10	−84.37	6	−269.28	3	1.13	3,608
1-50-100-25-15	−39.37	1,769	−273.87	412	49.66	3,601
5-5-20-8-5	−76.86	0	−199.40	1	0.00	3,205
5-10-100-15-10	−198.27	470	−109.05	30	3.47	3,601
5-10-50-15-10	−138.67	104	−163.85	12	9.31	3,601
10-5-20-8-5	−113.01	1	−126.65	1	0.00	3,605
10-10-100-15-10	−257.30	1,151	−62.30	98	0.89	3,601
10-10-50-15-10	−170.87	459	−105.63	43	7.14	3,601
Average	−120.58	398	−204.50	60	7.62	2,685

6.3. Sensitivity analysis and managerial insights

We now perform sensitivity analysis to derive important managerial insights. From Table 4, we observe that the more flexible the delivery time windows gets, the harder to solve the problem becomes. Also, as the number of products, periods, and customers increases, the problem becomes harder to be solved to optimality. Small instances with $P = 1$, $T = 5$, and $N_c = 20$ are easily solved to optimality, however, instances with only one product but $T > 5$ cannot be solved to optimality under the presence of any delivery time window.

This difficulty in solving the problem when delivery time windows exist shows two interesting aspects of the business problem. The first one is related to the potential cost saving if one is to properly exploit the added flexibility of time windows. This is evident since all solutions without time windows are still valid to the cases in which they are considered. However, to take advantage of such flexibility, using a tailored method seems necessary. As shown already, modeling the problem into a commercial solver or using a sequential method does not yield any good solutions. In fact, the quality of solutions degrades as the size of the problem and the added flexibility increase.

Figure 1 provides an overview on the comparison of our matheuristic and CPLEX for different delivery time windows. We compare the performance of both methods over the lower bound obtained by CPLEX. As observed in this figure, on average over all instances, our proposed algorithm works better when the delivery time window enlarges. The results reveal that for large instances our matheuristic outperforms the exact algorithm. The highest average improvement is obtained for $r = 5$. For all instances that could be solved to optimality by CPLEX, our algorithm also obtains the optimal solution.

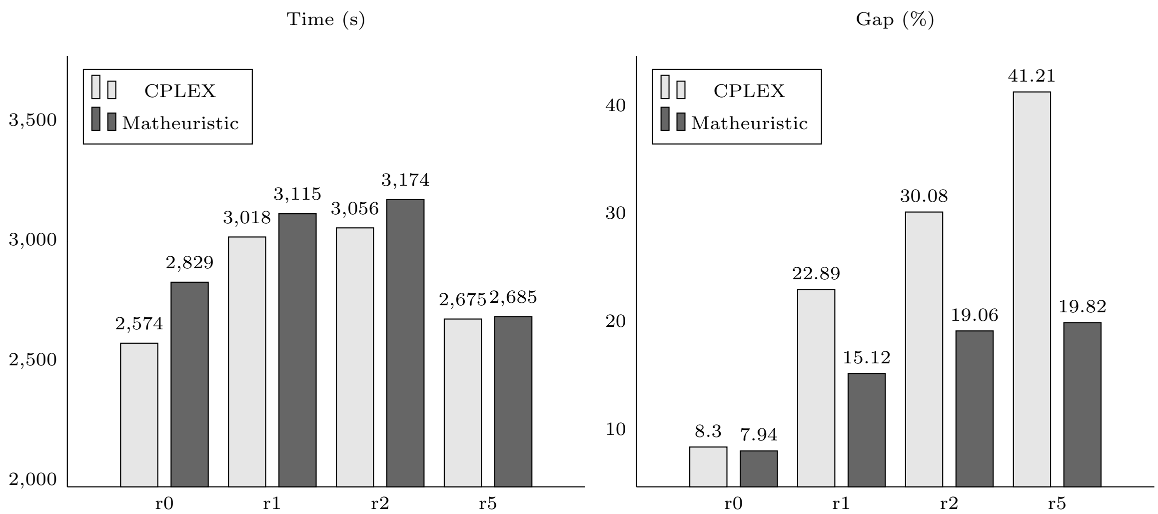

Figure 1: Comparison between time (s) and gap (%) of CPLEX and the proposed matheuristic

Considering the processing time, CPLEX performs slightly better, mainly because the iterative heuristic reaches the time limit to search the solution area, aiming to improve the solution obtained. However, as presented in Table 9, our algorithm takes on average less than 20 minutes to find its best solution, which is often better than the ones from the exact algorithm.

Table 9: Average time for the proposed method to obtain its best solution

Time window	$r = 0$	$r = 1$	$r = 2$	$r = 5$	Average
Average time (s)	1,160	1,159	1,022	909	1,063

Regarding our proposed matheuristic, Tables 5–8 also reveal that taking an integrative approach towards production, location, inventory, and distribution decisions can lead to enormous cost reductions. For all time windows, the average results obtained from the proposed method are always better than the sequential ones. It is interesting to note that on average the solutions obtained by Top-down procedure are lower than the ones from the Bottom-up approach; however, the Bottom-up procedure is much faster. As presented in Tables 5–8, the Bottom-up procedure generates better results, in less time, than the Top-down when $P > 1$ and $N_c > 20$.

As expected, applying the Equal power procedure, where each department of the company is focused only on its own decisions, results in not even one instance with a feasible solution. Comparing the solutions obtained by this procedure to the lower bounds of the exact algorithm, on average this infeasible solution from the Equal power procedure is 48.11% worse than the lower bound, which forgoes any hopes that this approach would yield any good solution. For this reason we do not provide detailed results from this method.

7. Conclusions

This paper investigates a challenging and practical problem of integrated production, location, inventory, and distribution, in which multiple products are produced over a discrete time horizon, stored at the DCs before being shipped to final customers. The paper contributes to the integrated optimization literature as it combines distinct features of delivery time windows, distribution with direct shipment, and dynamic location decisions. A state of the art commercial solver is able to find optimum solutions for very small instances of our problem, however, it does not prove optimality in a reasonable time for larger instances. To achieve better solutions in an acceptable computation time, we have proposed a mathuerisric algorithm. Several instances are generated and the solutions are compared to the optimal ones (if any) obtained by the exact method. On average the solutions obtained with our algorithm improve the ones from of the exact method by up to 49.66%, generally in only a third of the running time.

In this paper, we have also evaluated how a typical management in silos would perform, by deriving and implementing sequential solution methods. Our results confirm the cost benefits of the integrated approach towards decision making. Both Top-down and Bottom-up procedures perform worse than the exact methods as well as our proposed method. However, between these two procedures, the Bottom-up works better for instances with larger planning horizons and more products and customers, while Top-down is preferred when there is only one product and fewer than 20 customers.

Using our randomly generated instances validated by an industrial partner, we have shown the benefits of an integrated management, as opposed to the sequential one. Moreover, we have shown that for complex and rich integrated problems inspired by real-world cases, such as the one studied here, neither a hierarchical solution approach nor modeling and solving the problem by a commercial solver yield good solutions in a reasonable time. We have proposed a flexible and very powerful method which is capable of effectively handling all aspects of the problem in an efficient manner.

References

Adulyasak, Y., Cordeau, J.-F., Jans, R., 2015. The production routing problem: A review of formulations and solution algorithms. Computers & Operations Research 55, 141–152.

Akbalik, A., Penz, B., 2011. Comparison of just-in-time and time window delivery policies for a single-item capacitated lot sizing problem. International Journal of Production Research 49 (9), 2567–2585.

Amorim, P., Günther, H.-O., Almada-Lobo, B., 2012. Multi-objective integrated production and distribution planning of perishable products. International Journal of Production Economics 138 (1), 89–101.

Archetti, C., Bertazzi, L., Paletta, G., Speranza, M. G., 2011. Analysis of the maximum level policy in a production-distribution system. Computers & Operations Research 12 (38), 1731–1746.

Barbarosoğlu, G., Özgür, D., 1999. Hierarchical design of an integrated production and 2-echelon distribution system. European Journal of Operational Research 118 (3), 464–484.

Chen, Z.-L., 2010. Integrated production and outbound distribution scheduling: review and extensions. Operations Research 58 (1), 130–148.

Coelho, L. C., Cordeau, J.-F., Laporte, G., 2014. Thirty years of inventory routing. Transportation Science 48 (1), 1–19.

Cunha, J. O., Melo, R. A., 2016. On reformulations for the one-warehouse multi-retailer problem. Annals of Operations Research 238 (1-2), 99–122.

Darvish, M., Larrain, H., Coelho, L. C., 2016. A dynamic multi-plant lot-sizing and distribution problem. International Journal of Production Research 54 (22), 6707–6717.

De Matta, R., Hsu, V., Li, C.-L., 2015. Coordinated production and delivery for an exporter. IIE Transactions 47 (4), 373–391.

Ekşioğlu, S., Ekşioğlu, B., Romeijn, H., 2007. A lagrangean heuristic for integrated production and transportation planning problems in a dynamic, multi-item, two-layer supply chain. IIE Transactions 39 (2), 191–201.

Ekşioğlu, S. D., Romeijn, H. E., Pardalos, P. M., 2006. Cross-facility management of production and transportation planning problem. Computers & Operations Research 33 (11), 3231–3251.

Fahimnia, B., Farahani, R., Marian, R., Luong, L., 2013. A review and critique on integrated production–distribution planning models and techniques. Journal of Manufacturing Systems 32 (1), 1–19.

Fumero, F., Vercellis, C., 1999. Synchronized development of production, inventory and distribution schedules. Transportation Science 33 (3), 330–340.

Goldberg, A. V., 1997. An efficient implementation of a scaling minimum-cost flow algorithm. Journal of Algorithms 22 (1), 1–29.

Jayaraman, V., Pirkul, H., 2001. Planning and coordination of production and distribution facilities for multiple commodities. European Journal of Operational Research 133 (2), 394–408.

Jolayemi, J. K., Olorunniwo, F. O., 2004. A deterministic model for planning production quantities in a multi-plant, multi-warehouse environment with extensible capacities. International Journal of Production Economics 87 (2), 99–113.

Liang, Z., He, Y., Wu, T., Zhang, C., 2015. An informative column generation and decomposition method for a production planning and facility location problem. International Journal of Production Economics 170, 88–96.

Melo, R. A., Wolsey, L. A., 2012. MIP formulations and heuristics for two-level production-transportation problems. Computers & Operations Research 39 (11), 2776–2786.

Mula, J., Peidro, D., Díaz-Madroñero, M., Vicens, E., 2010. Mathematical programming models for supply chain production and transport planning. European Journal of Operational Research 204 (3), 377–390.

Mustafa Kamal, M., Irani, Z., 2014. Analysing supply chain integration through a systematic literature review: a normative perspective. Supply Chain Management: An International Journal 19 (5/6), 523–557.

Nezhad, A. M., Manzour, H., Salhi, S., 2013. Lagrangian relaxation heuristics for the uncapacitated single-source multi-product facility location problem. International Journal of Production Economics 145 (2), 713–723.

Park, Y., 2005. An integrated approach for production and distribution planning in supply chain management. International Journal of Production Research 43 (6), 1205–1224.

Power, D., 2005. Supply chain management integration and implementation: a literature review. Supply Chain Management: An International Journal 10 (4), 252–263.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborghood search heuristic for the pickup and delivery problem with time windows. Transportation Science 40 (4), 455–472.

Sambasivan, M., Schmidt, C. P., 2002. A heuristic procedure for solving multi-plant, mutli-item, mutli-period capacitated lot-sizing problems. Asia-Pacific Journal of Operational Research 19 (1), 87.

Sarmiento, A., Nagi, R., 1999. A review of integrated analysis of production–distribution systems. IIE Transactions 31 (11), 1061–1074.

Sharkey, T. C., Geunes, J., Edwin R., H., Shen, Z.-J. M., 2011. Exact algorithms for integrated facility location and production planning problems. Naval Research Logistics 58 (5), 419–436.

Speranza, M. G., 2016. Trends in transportation and logistics. European Journal of Operational Research, forthcoming.

Vogel, T., Almada-Lobo, B., Almeder, C., 2016. Integrated versus hierarchical approach to aggregate production planning and master production scheduling. OR Spectrum 39 (1), 193–229.